Improving Phoneme Recognition of Telephone Quality Speech

Qiang Huang and Stephen Cox

School of Computing Sciences,
University of East Anglia,
Norwich NR4 7TJ, U.K.
(hq|sjc}@cmp.uea.ac.uk

Abstract

There are some speech understanding applications in which training transcriptions are unavailable, and hence the vocabulary is unknown, but the task is to recognise key words and phrases within an utterance rather than to attempt a complete, accurate transcription. An example of such a task is call-routing, when transcriptions of training utterances (which are very expensive to produce) are unavailable. In such cases, phoneme rather than word recognition is appropriate. However, phoneme recognition of spontaneous speech spoken by a large multi-accent population over telephone connections is very inaccurate. To improve accuracy, we describe a technique in which we segment the waveform into subword-like units and use clustering and iteratively refined language model to correct the errors in the recognised phonemes. The results show a (46.76-28.06) reduction in phoneme error-rate.

1. Introduction

There are a number of useful applications that require recognition of spontaneous utterances made over telephone lines by a large population of speakers. One of these is *call-routing*, in which the task is to route automatically a call to one of a small number of "destinations". This is a difficult enough problem when the vocabulary is known *a priori*, but to get an estimate of the vocabulary used in any particular application (e.g. a department store, an insurance broker, a credit-card company) requires making transcriptions of example calls, which is a very expensive process. It is therefore much preferable to use unannotated speech for training, although we assume that each training utterance has at least been labelled with its "destination", a much less costly process than transcribing it. Assuming the vocabulary is not known at all, it is necessary to use phoneme recognition, and this is likely to be very inaccurate on such a signal. However, the task of routing utterances to a particular destination is made easier by the fact that the language used by the callers is usually highly formulaic (there are only a certain number of ways of e.g. asking for a balance!) and therefore constrained in its vocabulary and syntax. If this limitation can be exploited, it should be possible to achieve accurate call-routing without word-level transcriptions of utterances.

Previous research on "correction" of the phoneme sequence output by the recogniser has used N-best transcriptions [1 2], phone-lattices [3] Other research has focused on finding variable-length acoustic units by identifying multigrams in the recognised phoneme strings [4], but these approaches have used only clean speech provided by a small number of speakers, which guarantees a higher phoneme recognition accuracy than that obtainable in our application.

An obvious approach to this problem is to decode the training set utterances and then search for similarities in the decoded strings of utterances routed to the same destination, and differences in strings routed to different destinations, using a metric such as the Levenshtein Edit Distance. Use can also be made of phone lattices and N-best output to provide e.g. confidences on distances. However, the accuracy of our phoneme recogniser on this material is so low (28.1%) that the strings are too noisy to make this process useful. This low accuracy stems from the large number of pronunciation variations found in the spontaneous speech, and also from the fact that if the vocabulary is unknown, we cannot use a specific language model (in this case, a phonotactic model) to decode the speech and have to fall back on a generalised model.

In this paper, we describe a different approach to phoneme correction, in which we use both direct segmentation of the speech signal and information provided by the recognised phoneme strings. Information from both these sources is used to cluster the segments, and the recognised sequences of phonemes in the segments are then replaced by the appropriate cluster centres. The phonotactic language model (PLM) used by the recogniser can then be re-estimated on these revised recognised phoneme segments to get a new set of recognised phoneme strings, and the process iterated. The technique is described in detail in Section 3.

The organisation of this paper is as follows. In section 2, we describe the data used and the performance of speech recogniser. In section 3, we give the details of the technique and how it was applied. Section 4 presents and analyses the results and we end in Section 5 with a discussion and suggestions for future work.

2. Data and Recogniser

The application studied here was the enquiry-point for the store card for a large retail store. Customers were invited to call up the system and to make the kind of enquiry they would normally make when talking to an operator. Their calls were routed to 61 different destinations, although some destinations were used very infrequently. 15,000 utterances were available, but for these initial experiments, we used a subset totalling only 1042 utterances from three call-types. The call-types selected were "Paymentdate" (394 utterances) "Balance" (330 utterances) and "Replacecard" (318 utterances). These were selected for initial trials of the

algorithm because the salient phrases they contained used largely independent vocabularies.

Phoneme recognition of the input speech queries was performed using an HMM recogniser whose acoustic models had been trained on a large corpus of telephone speech and which had separate models for males and females. In addition, WSJ transcriptions were used to generate an initial 6-gram PLM, with standard backoff procedures when data is sparse. In this paper, we just used 1042 utterances from three call types to test our systems. The average length of an utterance is 8.01words. Because the data corpus we used are telephone quality spontaneous speech, phone error-rate is very high 72%, with about 34% substitutions, 33% deletions and 5% insertions.

3. Algorithm

Figure 1 gives an overview of our technique.

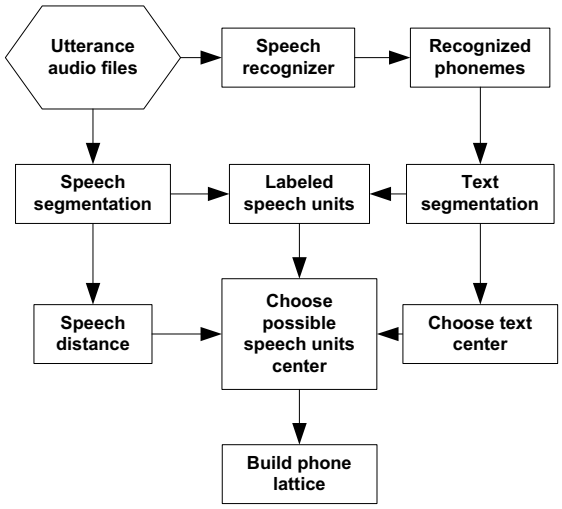

Figure 1: System Overview

The steps of our system are as follows. For each call-type:

1. Set the current phonotactic language model (PLM) to be a model constructed from some suitable speech material (e.g. Wall Street Journal (WSJ) transcriptions). We used a 6-gram model, with a standard backoff procedure.
2. Decode the speech from this call-type using a phoneme recogniser and the current PLM. Identify salient *phone sequences* (see Section 3.2).
3. Segment the speech into variable-length speech segments using standard speech signal-processing techniques (see Section 3.1). Each segment is labelled using the decoded phoneme string. In cases where the direct segmentation of the waveform does not match the segmentation provided by the recogniser, heuristic rules are used to make an approximate labelling. However, later iterations will increase the alignment of the two segmentations.
4. Identify salient *waveform segments*. This is done by comparing the label sequence for each waveform segment with the salient phone sequences. There are several waveform segments that match well to each different salient phone sequence.
5. Estimate a distance between each waveform segment and each salient waveform segment. We compute only the distance between each waveform segment and each salient segment in order to reduce greatly the computation time, which would be unfeasibly large if all segments were to be compared with each other.
6. Find the N segment waveforms in the data that are closest to each of the C salient waveforms. The set of waveforms closest to salient waveform number C_i constitutes the i^{th} cluster.
7. Correct the recognised phoneme strings by identifying the cluster closest to each string, and replacing the recognised string with the central string of the cluster.
8. Re-estimate the PLM using the corrected recognised phoneme strings. This iterative refinement of the PLM is similar to the technique used in [5].

We discuss some of these steps in more detail in the next sections.

3.1. Speech segmentation

Recently, research has moved from phone segmentation to word and syllable-based segmentation [6]. In order to get word and syllable-like speech units, techniques such as HMMs, neural networks, fuzzy logic and group delay function have been used to segment the speech. Although much of this work has been done on clean speech, in [7], experiments on the Switchboard corpus indicated that about 60% of word boundaries and 15% of phrase boundaries were found, which encourages us to use these techniques on telephone speech. In our system, we consider energy wrapping, zero-crossing rates and the group delay function. The energy envelope is used to find the silence gaps in the waveform and zero-crossing rates give a weak indication of differences in phones. If the short-term energy function is thought of as a magnitude spectrum, the group delay function can resolve the peaks and valleys of the spectrum well, but this tends to over-segment. Hence the group delay function is used in conjunction with the energy envelope and the zero-crossing to gives of estimates of word boundaries.

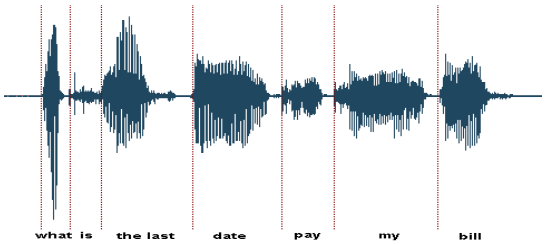

Figure 2 Segmented Waveform

Figure 2 shows the waveform of the utterance “What is the last date pay my bill” together with the segmentation produced by our segmentation algorithm. The waveform shows clear gaps between many of the words, and these have been correctly found by the algorithm. The phrase “the last” is not easy to segment by eye, and the algorithm has left this as a whole phrase, whilst finding the correct segmentation of the word “is” before this phrase.

CallType	*No. of words*	*No. of segmentations Found*	*No. of correct word segmentations*
Paymentdate	3098	4001	1682 (54.29%)
Balance	2211	2335	1273 (57.6%)
Replacecard	2932	3080	1667 (56.7%)

Table 1: Number of words and segmentations

Table 1 summarises the performance of the word segmentation algorithm. Because of the presence of line noise, breath noise, handset noise etc. there are more segments than the actual number of words in utterances. The segmentation was checked by listening to all 9416 segmentations. The third column shows that the accuracy is about 55%.

3.2. Choosing cluster centre of speech segmentations

Table 1 indicates that the number of speech waveform segments is very large, and it would be impossible to compare all pairs of segments in order to cluster them. To reduce the computational load, we attempted to identify a set of waveform segments as cluster centres. This is done as follows:

- Segment the recognised phoneme strings into n-grams ($n = 3, 4, \ldots, 7$)
- Choose a set of phoneme segments as salient segments. These are segments that occur more than a threshold T times.
- Label the segmented waveforms using the recognised phoneme strings.
- Compare the label sequences of the segmented waveforms with the salient phoneme sequences and choose as cluster centres the segmented speech waveforms whose labelled string is closest to the salient phoneme string.

3.3. Calculation of speech distance

The error rate of our phoneme recogniser is about 72%, which means that calculating distances between segments of speech using their label sequences is very difficult, because of the large number of errors on each segment. In any case, using labels derived from the complete utterance decoding is unlikely to give good results when comparing segments. Hence we prefer to compare the speech segments directly by calculating the distance between them using features derived from the speech signal. Using 8 PLP-Cepstra feature vectors was found to be better than using 12 MFCCs with deltas for this task. Dynamic time wrapping (DTW) was used by calculating to compare the similarity between two speech segments represented as PLP-Cepstra feature vectors.

$$D_{\vec{f}1\vec{f}2} = C_{12}\sum_{i,j}\frac{\vec{f}1_i \bullet \vec{f}2_j}{\left|\vec{f}1_i\right|\cdot\left|\vec{f}2_j\right|} \qquad (1)$$

where $\vec{f}1_i$ is the i^{th} feature vector of speech unit $\vec{f}1$ and $\vec{f}2_j$ is the j^{th} feature vector of speech unit $\vec{f}2$. $\frac{\vec{f}1_i \bullet \vec{f}2_j}{\left|\vec{f}1_i\right|\cdot\left|\vec{f}2_j\right|}$ gives the cosine of the angle between vectors $\vec{f}1$ and $\vec{f}2_j$. The weighting factor, C_{12}, is defined as:

$$C_{12} = 1/\left(L_{\vec{f}1}\cdot L_{\vec{f}2}\right) \qquad (2)$$

with $L_{\vec{f}1}$, $L_{\vec{f}2}$ are the length of speech segments $\vec{f}1$ and $\vec{f}2$ separately.

The distance between each salient waveform segment and all other waveform segments was estimated. For each salient waveform segment, the distances to the other segments were sorted, and the top M segments are used to build a finite state network that represents the labellings of these M segments. As an example of this process, the labels of the top 18 waveform segments that match to one of the salient waveforms in Figure 3

k I g~ I z
h e l I g~
p e m E m
g e m I g~
p e
k h I m I g~
I v I g~ I g~
k I l
p I n I g~
p I l er z
p I g~ E m
t I m E T
p I n E
h e l I
l I m E t
p e m E s
b t e n I g~
p I g~ m E n

Figure 3. The labels on 18 waveform segments of the word "payment"

Although the label sequences are very different, each waveform segment actually represents the word "payment". This shows that this clustering technique works well even when a large number of errors in the phoneme sequence output by the recogniser.

3.4. Optimal cluster center

A difficulty with the clustered segments (an example of which is shown in Figure 3) is that the large differences of labelling of these segments make it difficult to choose one as the cluster centre. To find the cluster centre, we segment each labelled phoneme string of a speech waveform into triphones, overlapping by two phones e.g. k I g~ I z → (k I g~) (I g~ I) (g~ I z). Let a segmented waveform sequence X be decomposed into a sequence of overlapping triphones $x_1, x_2, \cdots, x_n$. Each waveform sequence label is decomposed in this way and the number of occurrences of each triphone $O_1, O_2, \cdots, O_T$ is noted, where T is the number of triphones in the cluster. The cluster centre is chosen as the waveform segment whose label sequence,

when decomposed into a sequence of triphones, has the highest summed "score" as calculated using the counts $O_1, O_2, \cdots, O_T$.

Figure 4 shows an example of the scores for the top four waveforms in a cluster. This cluster represents the phrase "how much", which is transcribed as "*h aw m ^ tS*". The transcription "*h aw m ^ tS"* is the top-scoring label sequence and this sequence corresponds closely to the true transcription.

Label Sequence:

h aw m ^ tS	41
*k * m E*	18
*h aw m **	17
k m ^ tS	17

Figure 4. Scores for label sequences of waveforms within a cluster

3.5. Iterative language model

The idea of an iterative language model was described by Alshawi in [5]. His technique iterates phoneme recognition followed by re-estimation of the language model, with no speech segmentation and clustering as used in our technique. We compare this technique with our own technique in Section 4. Because results depend on the order of n-gram used, we ran an initial experiment to determine the best order of *n*-gram for our own PLM. Table 2 shows the results as *n* is increased from 2 to 6.

	2-gram	3-gram	4-gram	5-gram	6-gram
Phone Error-rate	73.92%	72.98%	72.33%	71.95%	71.94%

Table 2. Phone error rate with phone *n*-gram language model

Hence a 6-gram phonotactic language model was chosen.

4. Results

Figure 5 shows the phone accuracy for successive iterations of the technique described by Alshawi and our own technique. Both techniques show a sharp rise in accuracy after the first iteration but both also fall off a little after successive iterations. However, our own technique in which iteration of the language model is combined with segmentation and clustering of the waveforms is clearly superior to iteration of the language model alone.

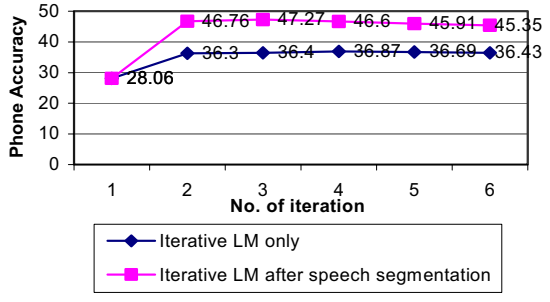

Figure 5. Phone Accuracy using iterative language model and speech segmentation

The phone accuracy improvement found in [5] is much greater than ours. This may be due to the number of words in utterance. The average length of an utterance in [5] is 3.95 words, whereas in our own work it is 8.01 words. The presence of more words in an utterance reduces the effects of context, which reduces the weight assigned to the most important *n*-gram in an utterance, and lowers the phone accuracy. The use of speech segmentation and clustering methods to correct errors boosts the performance over what is obtainable from merely iterating the language model

5. Conclusion

In this paper, we have presented a technique that uses speech segmentation, clustering and language model iteration to correct phone errors. The method was shown to work well on telephone quality spontaneous speech, raising the phoneme accuracy from 28.1% to 47.3%. The technique was compared with a similar technique that used only phoneme recognition followed by language model iteration, and was shown to give markedly superior performance on our data.

The goal of this work is to improve the accuracy of call-routing when no transcriptions of utterances are available. We are now testing this phone correction technique in a call-routing scenario to determine how successful it is.

6. Acknowledgment

We are grateful to Nuance Communications for providing the data for this study.

7. Reference

[1] Mehryar Mohri, Michael Riley, "An Efficient Algorithm for the N-Best-Strings Problem", in ICSLP 2002, Denver

[2] "Louis ten Bosch, Nick Cremelie, "Pronunciation Modeling and Lexical Adaption Using Small Training Set", in PMLA 2002.

[3] D. James, S. Young, "A Fast Lattice-based Approach to Vocabulary Independent Wordspotting", ICASSP 1994.

[4] Sabine Deligne, Frederic Bimbot, "Inference of variable-length linguistic and acoustic units by multigrams", in Speech Communication 23 (1997) pp.223-241.

[5] Hiyan Alshawi, "Effective Utterance Classification with Unsupervised Phonotactic Models", in Proceedings of HLT-NAACL 2003, pp. 1-7, Edmonton.

[6] T. Nagarajan, Hema Murthy, "Segmentation of Speech into Syllable-like units", EuroSpeech 2003, Geneva

[7] M. Ostendorf etc., "A Prosodically Labelled Databas of Spontaneous Speech", Proc. ISCA Tutorial and Research Workshop on Prosody in Speech Recognition and understanding, 2001.